TITLE: **METHOD FOR SERVER METAFARM CONFIGURATION**
         OPTIMIZATION

FIELD OF THE INVENTION:

        This invention relates to data processing
systems of the type which include a Server Metafarm that
includes one or more Server Farms that execute
5 application programs for multiple clients (users); and
more particularly, this invention relates to methods for
optimizing the "Server Metafarm size" by balancing a
number of Server Farms and availability requirements in
the   above   type   of   data   processing   systems.

CROSS-REFERENCES TO RELATED APPLICATIONS:

This application is related to co-pending applications, USSN 08/550,603 entitled "Estimator Program for Estimating the Availability of an Application Program That Runs in a Cluster of at Least Two Computers"; USSN 09/443,926, entitled "Method for Estimating the Availability of an Operating Server Farm", and USSN 09/474,706, entitled "Method for Server Farm Configuration Optimization" which applications are incorporated herein by reference.

BACKGROUND OF THE INVENTION:

The co-pending application "Estimator Program for Estimating the Availability of an Application Program That Runs in a Cluster of at Least Two Computers" referenced above involves an estimator program to perform method steps for estimating the availability of an application program that runs on any "server" in a cluster of at least two servers. By "availability of an application program" is meant the probability that at any particular time instance, at least one of the servers in a cluster (farm) will actually be servicing requests from external workstations able to use the application program.

In one embodiment, the so-called estimator program begins by receiving input parameters which include (i) multiple downtime periods for each computer in the cluster (farm) that occur at r spective frequ ncies due to various downtime sourc s, and (ii) an

application "failover" time period for switching the running of the application program from any one computer to another computer which is operable. From these input parameters, the described estimator program estimates

5   first and second annual stoppage times, then determines the availability of the application program on the cluster of computers which is derived from the sum of the first and second annual stoppage times.

Thus, as discussed, the estimator program of

10   the previously-described invention estimated a first annual stoppage time for the application program due solely to the concurrent stoppage of all of the computers, as a function of the ratio of a single computer virtual downtime period over the single computer

15   virtual time between stops. Then subsequently, the estimator program was used to estimate a second annual stoppage time for the application program, due solely to the switching for running the application program from one computer to another computer as a function of the

20   single virtual stoppage rate and the application failover time period. From this, the estimator program determined the availability of the application program on the cluster of computers by deriving the sum of the first and second annual stoppage times.

25   The estimator program method was based on the assumption that "application availability" was to be determined from four factors which were:

        (i)   single-server hardware reliability;

        (ii)   maintenanc , support, and service

30   strategies;

(iii) user application and environment;

(iv) failover or system reconnection mechanism and application recovery mechanism.

The prior estimation parameters which were
5   described in the co-pending application USSN 08/550,603 did not take into consideration the total number of operating Server Farm clients and the normal single server workload of users involved with each single server. Further, this earlier application did not
10   provide a recommendation or estimate regarding the number of servers required in the Server Farm (or cluster) which would meet the customers' performance and redundancy level requirements, nor did it establish an optimum farm configuration.
15   The method of the co-pending application USSN 09/443,926, entitled "Method for Estimating the Availability of an Operating Server Farm" extended the area of the original method application for Server Farms designed to serve user communities with a required
20   particular number of customers . This method involving the Server Farm size and availability calculations is based on (1) the single server parameters such as (a) the meantime to failure (MTTF), (b) the meantime to repair (MTTR), and (c) the single server application performance
25   benchmarks, and (2) individual customer preferential requirements, involving (a) the total number of Server Farm application users and (b) a desirable redundancy level.

This stimation method for availability uses
30   the following definition of Serv r Farm availability. This definition is th probability that a Server Farm

awk\appl\041479L.doc

provides access to applications and data for a particular minimum number of users. As soon as the Server Farm can not serve this particular minimum number of users, it is considered failed. When some of the users have lost

5 connections but can reconnect to other servers and continue to work, and the majority of users do not experience any interruptions in their work, the farm is not considered failed, if it can still serve this particular number of users.

10   A widely used approach to improve a system's availability beyond the availability of a single system is by using Server Farms with redundant servers. In this case, if one of the farm's servers fails, the "unlucky" users connected to this server will lose their

15 connections, but will have an opportunity to reconnect to other servers in the farm and get access to their applications and data. If all of the "unlucky" users get access to their applications and data, the farm is considered "available." If at least one of the "unlucky"

20 users fails to get access to his/her applications and data, it means that the Server Farm's redundancy was exhausted and the Server Farm is considered failed.

       The parameters for MTTF and MTTR can be estimated as indicated in the cited prior USSN 08/550,603

25 as a single computer virtual time between failures and a single computer virtual downtime period, respectively, for a particular application and user environment.

       Therefore, the availability estimation method of the prior application USSN 09/443,926 allows one to

30 estimate such parameters of th Server Farm as number of servers, Server Farm availability, and Server Farm

downtime, based on a set of input data. At the same tim ,
however, this method did not provide any recommendations
about optimum combinations of the Server Farm parameters
that can be chosen at the Server Farm planning or design
5 stage.

  The method of the co-pending application USSN
09/474,706, entitled "Method for Server Farm
Configuration Optimization" involving the Server Farm
size optimization is based on the input data that
10 includes single server parameters similar to the prior
application USSN 09/443,926 and at least two new extra
parameters which include: single server cost and the
downtime cost. Additionally, this method included newly
added steps of selecting an optimization parameter,
15 selecting an optimization criterion, and using an
optimization technique procedure to find the optimum
value of the optimization parameter. Therefore the
optimization method of the co-pending application USSN
09/474,706 allows one to find the optimum configuration
20 of the single Server Farm. At the same time, however,
this method had some technical limitations. The original
assumption, that the reliability of the workload
balancing mechanism is much higher than the reliability
of a single server running applications becomes
25 questionable if the Server Farm size exceeds 100-120
servers. This peculiar problem arises in the situation
where a very large number of users L, say 10,000, 50,000
or 100,000 users are involved.

  The presently described new method improves
30 Serv r Farm availability by using so called Metafarms
divid d into several Server Farms with a workload

balancing mechanism that distributes workload as between the Server Farms and as between the servers that make up the Server Farms. The Metafarm availability definition is similar to the Server Farm availability definition. The

5   Metafarm availability is the probability that a Metafarm provides access to applications and data for a particular minimum number of users. Therefore this is the probability that all Metafarm Server Farms are available.

The Metafarm design unlike Server Farm design

10   may involve a new parameter - the number of Server Farms of equal size that make up a Server Metafarm. The number of Server Farms can be used as one of the Server Metafarm optimization parameters.

While the present invention may be shown in a

15   preferential embodiment for a Metafarm that uses any workload balancing mechanism, it is not limited thereto, and can be used for any other data processing environment where the definition of the "Metafarm and Server Farm availability" can be applied.

20   Thus the object of the present invention is to provide a method for optimizing the Server Metafarm for delivery of service to this large number of users by balancing a number of Server Farms and their availability requirements. The method uses the fact that the

25   decomposition of the Metafarm into several Server Farms can increase the Server Metafarm availability. The method will generate an optimum recommendation for the selected set of input data. For example, the input data can include parameters of the particular server that is used

30   as a Metafarm building block and S rver Metafarm availability can be selected as optimization criterion.

awk\appl\041479L.doc

In this case, the optimum Server Metafarm configuration can include the values of the number of Server Farms of equal size that make up a Server Metafarm and the number of servers in each Server Farm.

5   The method of generating such optimum configurations for this type of network involving large number of users "L" is described and developed in the succeeding exposition hereinbelow.

SUMMARY OF THE INVENTION:

In accordance with the present invention, a novel estimator program performs method steps for the Server Metafarm optimization for a given particular large

5 number of Metafarm users/clients "L" by balancing a number of separate Server Farms and their availability requirements. By the optimization of the Metafarm is herein meant the process of finding the optimum values of the selected optimization parameters (such as a number of

10 Server Farms and/or a Redundancy Factor value) that deliver the optimum value (for the selected optimization criterion (such as Server Metafarm availability) and a given set of input data.

System performance in a Server Metafarm

15 computing environment involves a particular number of concurrent users with the minimum required application response time and reliable access to their applications and data. Server Metafarm availability is the probability that a Server Metafarm provides a required system

20 performance level. A Server Metafarm parameter that indirectly defines the Server Metafarm availability and performance is designated as a Redundancy Factor, that is a measure of the available system resources. The Redundancy Factor involves a difference between the

25 maximum required performance and the nominal performance as a percentage of the maximum performance.

In one particular embodiment, the method uses a simplified Server Metafarm availability model. The model uses Server Metafarm availability as an optimization

30 criterion. Th Redundancy Factor and the number of Server

awk\appl\041479L.doc

Farms that make up the Metafarm are used as optimization parameters.

The method uses the fact that the decomposition of the Metafarm into several Server Farms can increase 5 the server Metafarm availability.

BRIEF DESCRIPTION OF THE DRAWINGS:

Fig. 1 is a drawing of a data processing system which includes an application Server Metafarm that executes application programs for multiple clients 10 (users). This is an example of the typical system where the optimization method of balancing a number of Metafarm Server Farms and availability requirements can be applied;

Fig. 2 shows an example of the optimization 15 criterion as a function of one of the optimization parameters for a particular value of the second optimization parameter. Here, the optimization criterion is a Metafarm Availability and optimization parameters are a number of Metafarm Server Farms and a Redundancy 20 Factoris;

Fig. 3 shows an example of the optimization criterion as a function of one of the optimization parameters for a particular value of the second optimization parameter. Here, the optimization criterion 25 is a Metafarm MTTF and optimization parameters are a number of Metafarm Server Farms and a Redundancy Factor;

Fig. 4 is a flow chart which shows an estimator program, which performs method steps for estimating the optimum value of two of the possible optimization

parameters of an operating Server Metafarm designed to serve a particular overall number of clients "L".

GLOSSARY LIST OF RELEVANT ITEMS:

1.   AVAILABILITY: This is a measure of the readiness of the system and an application to deliver an expected service to the user with a required performance level.
5   It may be described as a percentage of time that a system and an application are running as distinguished from the system being down for maintenance or repairs.

2.   MEAN TIME TO FAILURE (MTTF): This is the average operating time between two failures, that can be
10   estimated as the total operating time divided by the number of failures.

3.   MEAN TIME TO REPAIR (MTTR): This is the average "downtime" in case of failure, that can be estimated as the total downtime divided by the number of failures.

15   4.   DOWNTIME: The downtime or repair time for a single application server is the time interval required to restore the server and system back to normal business operation. At the end of the repair period the applications running on the repaired server are available
20   to users. The downtime for a Server Farm is the time interval required to restore the nominal Server Farm performance.

5.   ESTIMATOR PROGRAM: This is a program which performs method steps for estimating system parameters such as the
25   availability of application programs running on any computer, server cluster, or server farm. This type of estimator program was the subject of co-pending

awk\appl\041479L.doc

applications USSN 550,603 and USSN 09/443,926, which are incorporated herein by reference.

6.   **SERVER METAFARM or METAFARM**:   This designates a group of identical Server Farms (see definition of the Server Farm below) that uses a workload balancing mechanism that distributes requests for services or applications to the available servers.

7.   **SERVER FARM**:   This designates a group of identical individual servers wherein each server can provide service to many single individual clients.   The Server Farm can run enterprise class client/server applications (SAP, PeopleSoft, Microsoft SQL) or applications that are traditionally run on a single workstation (Microsoft Office 97). The Server Farm usually uses a workload balancing mechanism that distributes requests for services or applications to the available servers.

8.   **REDUNDANCY FACTOR** (Rf):   This is a measure of the additional number of users that can be added to the nominal number of users per server without exceeding the maximum number of users per server (server performance benchmark maximum of users). It is a difference between maximum and nominal performance as a percentage of the maximum performance. The Redundancy Factor can be calculated as 100 percent minus a usage factor Uf.

9.   **SERVER FARM AVAILABILITY CALCULATOR**:   This is an estimator program which estimates the availability for the Server Farm.

10. THIN CLIENT SERVER FARM AVAILABILITY CALCULATOR: This is one of the examples of the SERVER FARM AVAILABILITY CALCULATORS. Because thin-client configurations are intended to make applications available to multiple users at the same time, this calculator calculates the availability of a specified number of instances of an application (not just a single instance) where each application instance is being run at the server, but all the user input response is taking place at the client terminal. In this scenario, downtime occurs whenever the number of available instances of the application drops below the required specified number of instances.

11. USAGE FACTOR (Uf): This is the ratio of the nominal number of users per server to the maximum number of users per server (server performance benchmark maximum of users) times 100 percent.

12. OPTIMIZATION CRITERION: This is a function that determines the value of one of the essential system attributes and must be minimized (or maximized) by variation of one or more system parameters that are chosen as OPTIMIZATION PARAMETERS. Each optimization parameter should have a predefined domain that defines the values that the optimization parameter may assume. The OPTIMIZATION CRITERION is a focus of an optimum system design or configuration. Examples of the optimization criteria are system performance, system availability, and cost of ownership.

awk\appl\041479L.doc

13. M_A = M tafarm Availability

14. M_{MTTF} = Metafarm Mean Time to Failure (MTTF)

15. K = Number of Server Farms in the Metafarm.

5 16. N = Number of Servers in the Server Farm

17. P = Maximum number of concurrent users per server or single server application performance benchmark

18. L = Total number of users being serviced

10   by the Metafarm, L=K*N*P

DESCRIPTION OF PREFERRED EMBODIMENT:

As seen in Fig. 1, there is illustrated a schematic drawing of a Server Metafarm 8 that consists of a number of Server Farms. These Server Farms will involve a series of Farms designated 10A, 10B, 10C,...,10K.

As a typical example, the Server Farm 10A involves a multiple number of servers designated A1, A2, A3,...,AN. Each of these servers has their own individual application programs respectively shown as 10PA1, 20PA2,...,NOPAN.

Likewise, each of the other Server Farms 10B, 10C, etc., has similar sets of hardware servers. For example, the Server Farm 10K of Fig. 1 has a set of servers designated K1, K2,...,KN, and these servers respectively have their own programs 10PK1, 20PK2,..., KOPKN.

A network 60 connects client terminals designated 1,2,...,L to enable connections to each server in the Server Metafarm. Additionally, each Server Farm has a database unit designated such as 12A for the Server Farm 10A. Likewise, the Server Farm 10K has a database unit designated 12K.

The network bus 60 of Fig. 1 provides connections from the multiple number "L" of client terminals shown as client terminal 1, 2, . . . L.

It should be understood here that this type of situation with the use of a Metafarm involves a large number of cli nt users L, such as 10,000, 50,000, or 100,000 and so on. This is the r ason that a Server

M tafarm, such as Server Metafarm 8, requires multiple numbers of Server Farms, such as 10A, 10B, 10C,...,10K.

One of the major problems that arises in optimizing a configuration for a given total number of
5 users L, is to decide how many Server Farms K of equal size N should be used to constitute the entire Server Metafarm, where N = the number of Servers in a Server Farm.

As was illustrated in USSN 09/474,706, the
10 Redundancy Factor is a key parameter in estimating Server Farm availability since it defines Server Farm size. The Redundancy Factor was used as one of the optimization parameters for the optimum Server Farm configuration. The definition of the Redundancy Factor can be applied to
15 Metafarms also. For example, if a Metafarm should serve a large number of users where L=30,000 (Metafarm users) and each of the servers that make up this Metafarm can serve P=100 users, then a Metafarm size K=1 (it is just one farm) and a Server Farm size N (that is a minimum number
20 of servers that can serve L users) is equal to N=30,000/100=300 servers. This Metafarm does not have any redundant servers. According to the Metafarm availability definition, if any of the 300 servers fails, the Metafarm will be considered failed. To improve the Metafarm
25 availability, redundant servers can be added. In case of the Redundancy Factor Rf=10%, the Metafarm resources to serve about 3,000 additional users are considered redundant. In terms of servers, it is about 30 additional servers. (30 is 10% of 300). Unfortunately, the original
30 assumption (made in USSN 09/474,706) that th reliability of the Server Farm workload balancing mechanism is much

higher than the reliability of a single serv r running applications becomes questionable if the Server Farm size exceeds 100-120 servers. This is why the novel method described here uses the decomposition of the Metafarm

5   into several Server Farms. The Metafarm can be divided into different number of the identical Server Farms with the same number of servers per Server Farm. The redundant servers can then be distributed evenly between Server Farms.

10   The goal of the reliable Metafarm design is to find a configuration with a minimum number of Metafarm servers (optimum configuration) that meets two major requirements:

        1) performance requirement: to serve a given

15   large number of users L;

        2) availability requirement: provide a given Metafarm Availability value M_A or a given Metafarm MTTF value M_{MTTF}.

        Two optimization parameters that define the

20   optimum configuration are the number of Server Farms K that make up the Metafarm and the Redundancy Factor that defines the overall number of servers.

        Fig. 2 illustrates an example of the Metafarm optimization. Fig. 2 is a graphical plot of availability

25   against the number "K" of Server Farms for a specified range of Redundancy Factors. The data used for optimization include the following:

        a) performance requirement: provide services for L=30,000 users;

30

b)   availability r quirement: provide M tafarm
Availability $M_A=0.99999$;

c)   single server that runs application programs
MTTF=1,500 hours;

5   d)   single server that runs application programs
MTTR=6 hours;

e)   single server that runs application programs
performance benchmark P=100 concurrent users;

f)   Redundancy Factor domain is an interval

10   (10%, 14%);

g)   Number of Server Farms is a natural number
1, 2, . . . 10.

There are three configurations on Fig. 2 with
availability M_A that is greater or equal 0.99999. Two
15   configurations are for a Redundancy Factor that is equal
to Rf=14% (where the First configuration includes 345
servers that are distributed between 3 server farms. Each
server farm contains 115 servers where 14 servers are
considered redundant servers. The second configuration
20   includes 344 servers that are distributed between 4
server farms. Each server farm contains 86 servers where
10 servers are considered redundant servers). The third
configuration is for a Redundancy Factor that is equal to
Rf=13%. (This third configuration includes 342 servers
25   that are distributed between 3 server farms. Each server
farm contains 114 servers where 13 servers are considered
redundant servers). It is obvious, that the configuration
with the smaller value of the Redundancy Factor Rf=13%
and thr Server Farms is the optimum configuration for

this example, because it requires less servers than th two other configurations.

Fig. 3 illustrates the same example of the Metafarm optimization where the availability requirement
5 is given in the form of the Metafarm MTTF value where M_{MTTF} is equal to 600,000 hours. Again, there are three configurations on Fig. 3 with Metafarm MTTF that is greater than or equal to 600,000 hours. Two configurations are for a Redundancy Factor Rf that is
10 equal to 14% and one configuration is for a Redundancy Factor Rf that is equal to 13%. The value of the Metafarm MTTF for 3 Server Farms and Redundancy Factor Rf=14% is equal to 1,843,768 hours and is not shown in Fig. 3 due to limited graph size. The optimum configuration is the
15 same as for the availability requirement in the form of the Metafarm Availability value that is equal to 0.99999: so that 3 Server Farms with the Redundancy Factor Rf of 13% is seen to have the least number of servers that deliver MTTF that is greater than the required goal
20 (600,000 hours).

Now, in accordance with the present invention, steps are provided for optimization of a Server Metafarm designed to serve a particular large number of users "L". These steps will be described in conjunction with Fig. 4
25 that shows an estimator program, which performs method steps for estimating parameters of the optimum operating Server Metafarm.

In step A of Fig. 4, requests for the following input parameters are displayed: (1) performance
30 requirement in the form of requir d large number of users "L" for utilizing said Server Metafarm, (2) th maximum

awk\appl\041479L.doc

single-s rv r workload "P" of concurrent users (the performance benchmark), (3) the Mean Time To Repair for a single server, and (4) Mean Time To Failure for a single server.

5   In step B of Fig. 4, the values of the requested input parameters are entered on the computer monitor by means of the keyboard.

In step C of Fig. 4, an optimization parameter (parameters) and its (their) domain is (are) selected.
10   For example, in Fig. 2 and 3, the selected optimization parameters are the Number of Server farms K and the Redundancy Factor Rf. The Number of Server Farms is a natural number 1, 2, . . . 10. The Redundancy Factor domain is an interval (10%,14%) that is the interval
15   between 0 and 100 percent.

In step D of Fig. 4, an optimization criterion is selected. In Fig. 2, for example, the selected optimization criterion is the Redundancy Factor that should be minimized and, additionally, the selected
20   constraint in the form of the minimum required Server Metafarm Availability value is the Metafarm Availability goal M_A that should be greater or equal to 0.99999. In Fig. 3, for another example, the selected optimization criterion is the Redundancy Factor that should be
25   minimized and, additionally, the selected constraint in the form of the minimum required Server Metafarm MTTF value is the Metafarm MTTF goal M_{MTTF} that should be greater than or equal to 600,000 hours.

In step E of Fig. 4, optimization of the
30   optimization criterion selected at step D occurs by one of th known optimization techniques described in the

awk\appl\041479L.doc

books: *Practical Optimization* by Philip Gill, Academic Press, 1981 and/or *Engineering Optimization: Methods and Applications* by G. V. Reklaitis and others, John Wiley & Sons, 1983. Fig. 2 and 3 illustrate one of the simplest

5   optimization techniques by the plotting of the graph for a set of values of the optimization parameters. Particularly, the used values of the Number of Server Farms K that make up the Metafarm are natural numbers 1, 2, . . . 10 and the used values of the Redundancy Factor

10   are from 10% to 14% percent with steps of 1%: as for example, 10, 11, . . . 14%. If the accuracy of the calculation is not sufficient, then the value of a step can be reduced from 1% to 0.5%, etc.

In step F of Fig. 4, the optimum values of the

15   optimization parameters are displayed. For example, in Fig. 2 and 3, the optimum value of the Number of Server Farms K that make up the Metafarm is 3 Server Farms and the optimum value of the Redundancy Factor is 13% which corresponds to the Metafarm with 342 servers.

20   Described herein has been an optimization method and an estimator program for a Server Metafarm designed to serve a particular large number of clients L.

While a preferred implementation of the invention has been described, it should be understood

25   that other implementations may be used which are still encompassed by the attached claims.

awk\appl\041479L.doc